CURRENT LISTING OF CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1   1.-3.   (Cancelled)

1   4.   (Previously Presented) A method comprising:

2   receiving a first tuple into a base relation at a first node of a parallel

3 database system having plural nodes, wherein the first tuple comprises a join attribute and

4 the base relation is partitioned across the nodes according to an attribute different from

5 the join attribute;

6   storing the first tuple in an auxiliary relation at a second node of the

7 parallel database system, wherein the auxiliary relation is partitioned across the nodes of

8 the database system according to the join attribute;

9   identifying second tuples of a second relation;

10   joining the first tuple with the second tuples based on the join attribute to

11 produce join results; and

12   storing the join results in a join view.

1   5.-6. (Cancelled)

1   7.   (Previously Presented) The method of claim 4, wherein storing the first

2 tuple in an auxiliary relation at a second node comprises:

3   determining that a join view definition excludes an attribute of the first

4 tuple; and

5   not storing the excluded attribute in the auxiliary relation.

1   8.   (Previously Presented) The method of claim 4, further comprising:

2   receiving a third tuple into the base relation;

3   determining that a join view definition includes a condition on one of the

4   attributes of the third tuple; and

5   determining that the condition is not met by the one of the attributes of the

6   third tuple; and

7   not storing the second tuple in the auxiliary relation.

1   9. - 15. (Cancelled)

1   16.   (Previously Presented) An article comprising a medium storing

2   instructions for enabling a processor-based system having plural nodes to:

3   store a join view to store join results of a join of at least first and second

4   base relations based on a join condition including a first attribute of the first base relation

5   and a second attribute of the second base relation;

6   receive a first tuple into the first base relation at a first node, wherein the

7   first tuple comprises the first attribute and the first base relation is partitioned across the

8   plural nodes according to an attribute other than the first attribute;

9   store the first tuple in a first auxiliary relation at a second node, wherein

10   the first auxiliary relation is partitioned across the plural nodes according to the first

11   attribute;

12   identify second tuples of the second base relation; and

13   join the first tuple with the second tuples to produce join results for

14   updating the join view.

1   17.   (Previously Presented) The article of claim 16, further storing instructions

2   for enabling the processor-based system to:

3   compare the second attributes of the second tuples with the first attribute

4   of the first tuple to produce the join results for updating the join view.

1   18.   (Previously Presented) The article of claim 16, further storing instructions

2   for enabling the processor-based system to:

3   determine that a join view definition excludes an attribute of the first

4   tuple; and

5   not store the excluded attribute in the first auxiliary relation.

1   19.   (Previously Presented) The article of claim 16, further storing instructions

2   for enabling the processor-based system to:

3   determine that a join view definition includes a condition on one of the

4   attributes of the first base relation; and

5   identify the one attribute in a received third tuple;

6   determine that the condition is not met by the received third tuple; and

7   not store the received third tuple in the first auxiliary relation.

1   20. - 32.   (Cancelled)

1   33.   (Previously Presented) The method of claim 4, wherein joining the first

2   tuple with the second tuples is performed at the second node.

1   34.-40.   (Cancelled)

1   41.   (Previously Presented) A database system comprising:

2   storage modules to store base relations and at least a first auxiliary relation

3   corresponding to a first one of the base relations, the first auxiliary relation containing

4   one or more attributes of the first base relation, the first auxiliary relation partitioned

5   across the storage modules differently than the first base relation, the storage modules

6   further to store a join view based on a join of the base relations; and

7   a controller adapted to update the join view using at least the first auxiliary

8   relation,

9   the join view to store results of a join of the base relations based on a query

10   containing a select clause and a join condition, the select clause specifying one or more

11   attributes of the first base relation,

12   the controller to store the one or more attributes of the first base relation

13   specified by the select clause in the first auxiliary relation, and the controller to not store

14   other attributes of the first base relation not specified by the select clause in the first

15   auxiliary relation.

1   42.   (Previously Presented) A database system comprising:

2   storage modules to store base relations and at least a first auxiliary relation

3   corresponding to a first one of the base relations, the first auxiliary relation containing

4   one or more attributes of the first base relation, the first auxiliary relation partitioned

5   across the storage modules differently than the first base relation, the storage modules

6   further to store a join view based on a join of the base relations; and

7   a controller adapted to update the join view using at least the first auxiliary

8   relation,

9   the join view to store results of a join of at least the first base relation and

10   a second base relation based on a join condition including a first attribute of the first base

11   relation and a second attribute of the second base relation, wherein the first attribute is a

12   key of the first base relation and the second attribute is a foreign key of the second base

13   relation that references the first attribute,

14   the controller to, in response to detecting that the first attribute is a key of

15   the first base relation and that the second attribute is a foreign key of the second base

16   relation that references the first attribute, create the first auxiliary relation to store the one

17   or more tuples of the first base relation but to not create a second auxiliary relation to

18   store tuples of the second base relation.

5

1   43.   (Previously Presented) The article of claim 16, wherein the processor-

2   based system comprises plural storage modules, the first base relation partitioned across

3   the storage modules according to the attribute other than the first attribute, and

4   the first auxiliary relation partitioned across the storage modules according

5   to the first attribute.

1   44.   (Previously Presented) The article of claim 43, wherein the plural nodes

2   contain the storage modules and plural processors, wherein the instructions when

3   executed cause the processor-based system to further distribute the first tuple from the

4   first node to the second node.

1   45.   (Previously Presented) The article of claim 44, wherein the instructions

2   when executed cause the processor-based system to further:

3   store a second auxiliary relation containing one or more tuples of the

4   second base relation;

5   partition the second auxiliary relation across the storage modules

6   according to the second attribute; and

7   partition the second base relation across the storage modules according to

8   an attribute of the second base relation other than the second attribute.

1   46.   (Previously Presented) The article of claim 45, wherein identifying the

2   tuples of the second base relation comprises identifying the tuples of the second auxiliary

3   base relation, and wherein joining the first tuple with the second tuples comprises joining

4   the first tuple with the second tuples of the second auxiliary relation.

1   47.   (Previously Presented) The article of claim 16, wherein the join view

2   stores join results of the join of the at least first and second base relations based on a

3   query containing the join condition, wherein the instructions when executed cause the

4   processor-based system to further determine whether the query specifies one or more

5   elements that enable storage of less than the entire first base relation in the first auxiliary

6   relation.

1   48.   (Previously Presented) The article of claim 47, wherein the first auxiliary

2   relation stores the entire first base relation in response to determining the one or more

3   elements do not exist.

1   49.   (Previously Presented) The article of claim 48, wherein the one or more

2   elements comprise a selection condition in a WHERE clause of the query.

1   50.   (Previously Presented) The article of claim 48, wherein the one or more

2   elements comprise less than all of the attributes of the first base relation specified by a

3   select clause in the query.